exception handling

import java.util.Scanner;

public class ExceptionHandling {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

switch(choice) {case 1: try {

System.out.println("Enter numerator and denominator:");

int numerator = scanner.nextInt();

int denominator = scanner.nextInt();

int result = numerator / denominator;

System.out.println("Result: " + result);

} catch (ArithmeticException e) {

System.out.println("Error: Division by zero is not allowed.");}

break;case 2:try {int[] array = {1, 2, 3};

System.out.println("Enter index:");

int index = scanner.nextInt();

System.out.println("Element at index " + index + ": " + array[index]);

} catch (ArrayIndexOutOfBoundsException e) {

System.out.println("Error: Array index is out of bounds.");}

break;case 3:try {

System.out.println("Enter a number:");

String input = scanner.next();

int number = Integer.parseInt(input);

System.out.println("Number: " + number);

} catch (NumberFormatException e) {

System.out.println("Error: Invalid number format.");break;

case 4: try {

System.out.println("Enter a string:");

String str = scanner.next();

System.out.println("Enter index:");

int index = scanner.nextInt();

char ch = str.charAt(index);

System.out.println("Character at index " + index + ": " + ch);

} catch (StringIndexOutOfBoundsException e) {

System.out.println("Error: String index is out of bounds.");

}break;case 5:try {String str = null;

System.out.println("Length of the string: " + str.length());

} catch (NullPointerException e) {

System.out.println("Error: Null pointer exception.");

}break;default:

System.out.println("Invalid choice.");}

scanner.close();}}

MULTITHREAD PROGRAMMMMM

import java.util.Random;

class GenerateNumber implements Runnable {

public void run() {

Random random = new Random();

while (true) {

int number = random.nextInt(100);

System.out.println("Generated Number: " + number);

if (number % 2 == 0) {

new Thread(new SquareNumber(number)).start();

} else {

new Thread(new CubeNumber(number)).start();

}try {

Thread.sleep(1000); // Sleep for 1 second

} catch (InterruptedException e) {

System.out.println(e);

}}}}class SquareNumber implements Runnable {

private int number;

SquareNumber(int number) {

this.number = number;}

public void run() {

System.out.println("Square of " + number + " is " + (number \* number));

}}class CubeNumber implements Runnable {

private int number;

CubeNumber(int number) {

this.number = number;

}public void run() {

System.out.println("Cube of " + number + " is " + (number \* number \* number));

}}public class MultiThreadedApp {

public static void main(String[] args) {

Thread generateThread = new Thread(new GenerateNumber());

generateThread.start();

}}

SORTING LAST WALA(12)

import java.util.Arrays;

class Sort {

public <T extends Comparable<T>> void Arrange(T[] array) {

Arrays.sort(array);

}public <T> void Display(T[] array) {

for (T element : array) {

System.out.print(element + " ");

}System.out.println();}}

public class GenericSortExample {

public static void main(String[] args) {

Sort sorter = new Sort();

Integer[] intArray = {5, 3, 8, 1, 9};

System.out.println("Original Integer Array: ");

sorter.Display(intArray);

sorter.Arrange(intArray);

System.out.println("Sorted Integer Array: ");

sorter.Display(intArray);

String[] strArray = {"Banana", "Apple", "Cherry", "Date"};

System.out.println("\nOriginal String Array: ");

sorter.Display(strArray);

sorter.Arrange(strArray);

System.out.println("Sorted String Array: ");

sorter.Display(strArray);

Double[] doubleArray = {2.5, 3.7, 1.2, 4.8, 0.9};

System.out.println("\nOriginal Double Array: ");

sorter.Display(doubleArray);

sorter.Arrange(doubleArray);

System.out.println("Sorted Double Array: ");

sorter.Display(doubleArray);}}

7th import java.util.Scanner;

class circle {double radius;

String color;circle() {

radius = 1.0;color = "blue";

}circle(double radius) {

this.radius = radius;color = "blue";

}circle(double radius, String color) {

this.radius = radius;this.color = color;

}double getarea() {return Math.PI \* radius \* radius;

}double getradius() {return radius;

}String getcolor() {return color;

}}class cylinder extends circle {

double height;double getheight() {

return height;}cylinder() {super();

height = 2.0;}cylinder(double height) {

super();this.height = height;}

cylinder(double height, double radius) {

super(radius);this.height = height;}

cylinder(double height, double radius, String color) {

super(radius, color);this.height = height;}

double getarea() {return (2 \* Math.PI \* radius \* height)

+ (2 \* Math.PI \* radius \* radius);}

double getvolume() {return super.getarea() \* height;

}void display() {System.out.println("Radius is " +

super.radius + ", Height is " + height + ", Color is "

+ super.color + ", Area is " + getarea() + ", Volume is " + getvolume());}

void check(cylinder c1, cylinder c2, int i, int j) {

if (c1.radius == c2.radius && c1.height == c2.height &&

c1.color.equalsIgnoreCase(c2.color)) {System.out.println

("The cylinders " + (i + 1) + " and " + (j + 1) + " are similar");}}}

public class Main {public static void main(String[] args) {

Scanner s = new Scanner(System.in);cylinder[] c = new cylinder[4];

c[0] = new cylinder();c[1] = new cylinder(3.0);

c[2] = new cylinder(3.0, 4.0);c[3] = new cylinder(3.0, 4.0, "Green");

System.out.println("Enter the details of cylinder 4 (height, radius, and color):");

double h = s.nextDouble();double r = s.nextDouble();String str = s.next();

c[3] = new cylinder(h, r, str);for (int i = 0; i < 4; i++) {

System.out.print("The dimensions of cylinder " + (i + 1) + " is ");

c[i].display();}for (int i = 0; i < 4; i++) {for (int j = i + 1; j < 4; j++) {

c[i].check(c[i], c[j], i, j);}}}}

8th one

class TestAccountInterface {

public static void main(String[] args) {

IAccount account = new HDFCAccount();

System.out.println("Transacting using HDFC Account");

transactOnAccount(account);System.out.println();

account = new StateBankAccount();

System.out.println("Transacting using State Bank Account");

transactOnAccount(account);}

public static void transactOnAccount(IAccount account) {

System.out.println();account.deposit(1000.0);

printBalance("depositing 1,000.0", account);

account.withdraw(2500.0);

printBalance("withdrawing 2,500.0", account);

account.deposit(4100.0);

printBalance("depositing 4,100.0", account);

account.withdraw(5000.0);

printBalance("withdrawing 5,000.0", account);

System.out.println();

}public static void printBalance(String message, IAccount account) {

System.out.println("The balance after " + message + " is " + account.getBalance() + ".");

}}interface IAccount {double getBalance();

void deposit(double amount);void withdraw(double amount);

}class HDFCAccount implements IAccount {

private double totalDeposits;private double totalWithdrawals;

@Override public double getBalance() {

return totalDeposits - totalWithdrawals;

}@Override public void deposit(double amount) {

if (amount > 0) {totalDeposits += amount;} else {

System.out.println("Invalid deposit amount. Please enter a positive value.");

}}@Override public void withdraw(double amount) {

if (amount > 0) {totalWithdrawals += amount;} else {

System.out.println("Invalid withdrawal amount. Please enter a positive value.");

}}}class StateBankAccount implements IAccount {private double balance;

@Override public double getBalance() {return balance;}

@Override public void deposit(double amount) {if (amount > 0) {

balance += amount;} else {

System.out.println("Invalid deposit amount. Please enter a positive value.");

}}@Override

public void withdraw(double amount) {if (amount > 0) {

balance -= amount;} else {

System.out.println("Invalid withdrawal amount. Please enter a positive value.");}}}

package wala

cie package> student class>

// File: CIE/Student.java

package CIE;public class Student {

public String usn;public String name;

public int sem;}

cie>internals class>// File: CIE/Internals.java

package CIE;public class Internals extends Student {

public int[] internalMarks = new int[6];

public Internals(String usn, String name, int sem, int[] internalMarks) {

this.usn = usn;this.name = name;this.sem = sem;

this.internalMarks = internalMarks;}}

see packaage>external class > // File: SEE/External.java

package SEE;import CIE.Student;

public class External extends Student {

public int[] seeMarks = new int[6];

public External(String usn, String name, int sem, int[] seeMarks) {

this.usn = usn;this.name = name;this.sem = sem;

this.seeMarks = seeMarks;}}

main.java>// File: Main.java

import CIE.Internals;import SEE.External;

public class Main {

public static void main(String[] args) {

int N = 5; Internals[] internalStudents = new Internals[N];

External[] externalStudents = new External[N];

for (int i = 0; i < N; i++) {internalStudents[i] = new Internals

("USN" + (i + 1), "Student" + (i + 1), 3, new int[]

{80, 85, 75, 90, 88, 92});externalStudents[i] = new External

("USN" + (i + 1), "Student" + (i + 1), 3, new int[]{70, 75, 65, 80, 78, 82});

}for (int i = 0; i < N; i++) {

System.out.println("Student: " + internalStudents[i].name);

System.out.println("USN: " + internalStudents[i].usn);

System.out.println("Semester: " + internalStudents[i].sem);

int totalMarks = 0;for (int j = 0; j < 6; j++) {

int finalMarks = internalStudents[i].internalMarks[j] + externalStudents[i].seeMarks[j];

totalMarks += finalMarks;

System.out.println("Course " + (j + 1) + " Final Marks: " + finalMarks);}

System.out.println("Total Marks: " + totalMarks + "\n");}}}